# EC447

Fall 2016

Boston University

# Course Components

- Learn to program in C# using the .NET framework
- Learn to develop Windows applications using the .NET framework class library
- Investigate how the technologies work
- Complete challenging lab assignments

# DreamSpark

- You will have access to the DreamSpark software download site (ELMS) which gives you access to thousands of dollars in development software.
- You must abide by the license agreement I will discuss.
- The URL for  theELMS site is https://e5.onthehub.com/WebStore/ProductsByMajorVersionList.aspx?ws=5ab2d1df-c998-e311-93fa-b8ca3a5db7a1&vsro=8
- Log in with your BU Kerberos credentials.
- This site is now managed by the College of Engineering and I can not provide any support. Contact ENG IT at enghelp@bu.edu for any issues you may have. Please do not e-mail me.

# Visual Studio 2015

- This course requires VS2015. If you have an older version install the newer version.
- Start on this right away as installation takes time.
- If you have never used Visual Studio there is a distinct learning curve. I will not be teaching the basics and fully expect that you should be able to learn it on your own.
- Don’t waste the next week. Get the job done now!

# Visual Studio 2015

- If you are short on disk space do not install Visual C++ or Visual Basic etc. (do a custom install)
- By default the help files are accessed over the Internet. You can also install the documentation locally.
- Choose at least the VS 2015 and .NET 4 documentation.
- You can go to the help file manager later to add additional content.

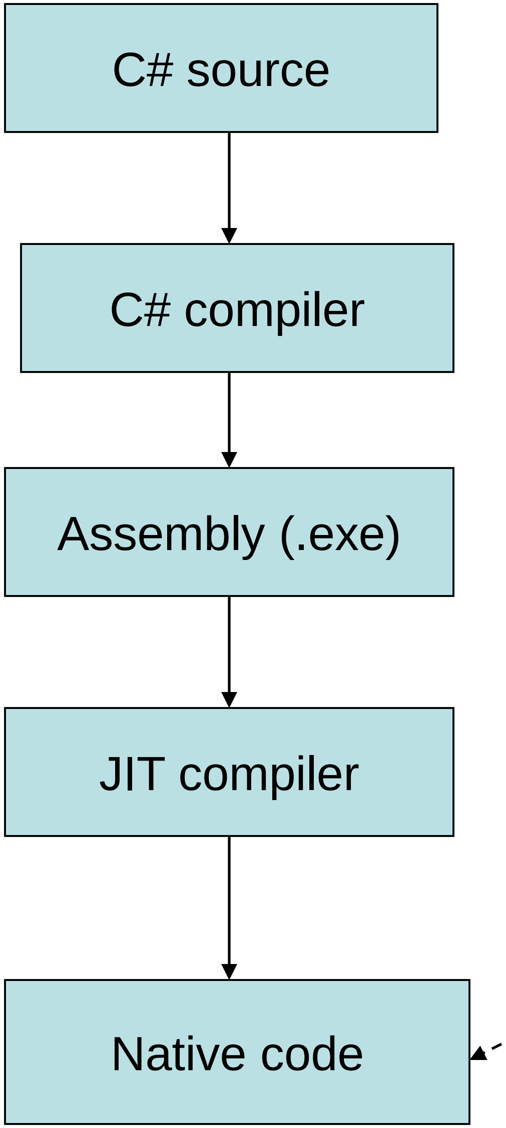
# The .NET Technology Elements

- Supports multiple programming languages
- Provides a common intermediate language and JIT compilation
- Supports *managed* code for reliability
- Makes extensive use of a *framework class library*
- Include client and server side technologies
- Is highly object oriented
- Uses standards such as XML

# The .NET Framework

- The platform for building and running applications
- Common language runtime, CLR
  - Managed applications
  - Garbage collection
- Framework class library, FCL
- Common intermediate language, CIL (MSIL)
- Metadata (eliminates the need for header files)
- Just in time compiling, JIT

Sequence of steps to compile and execute a C# program



C# source

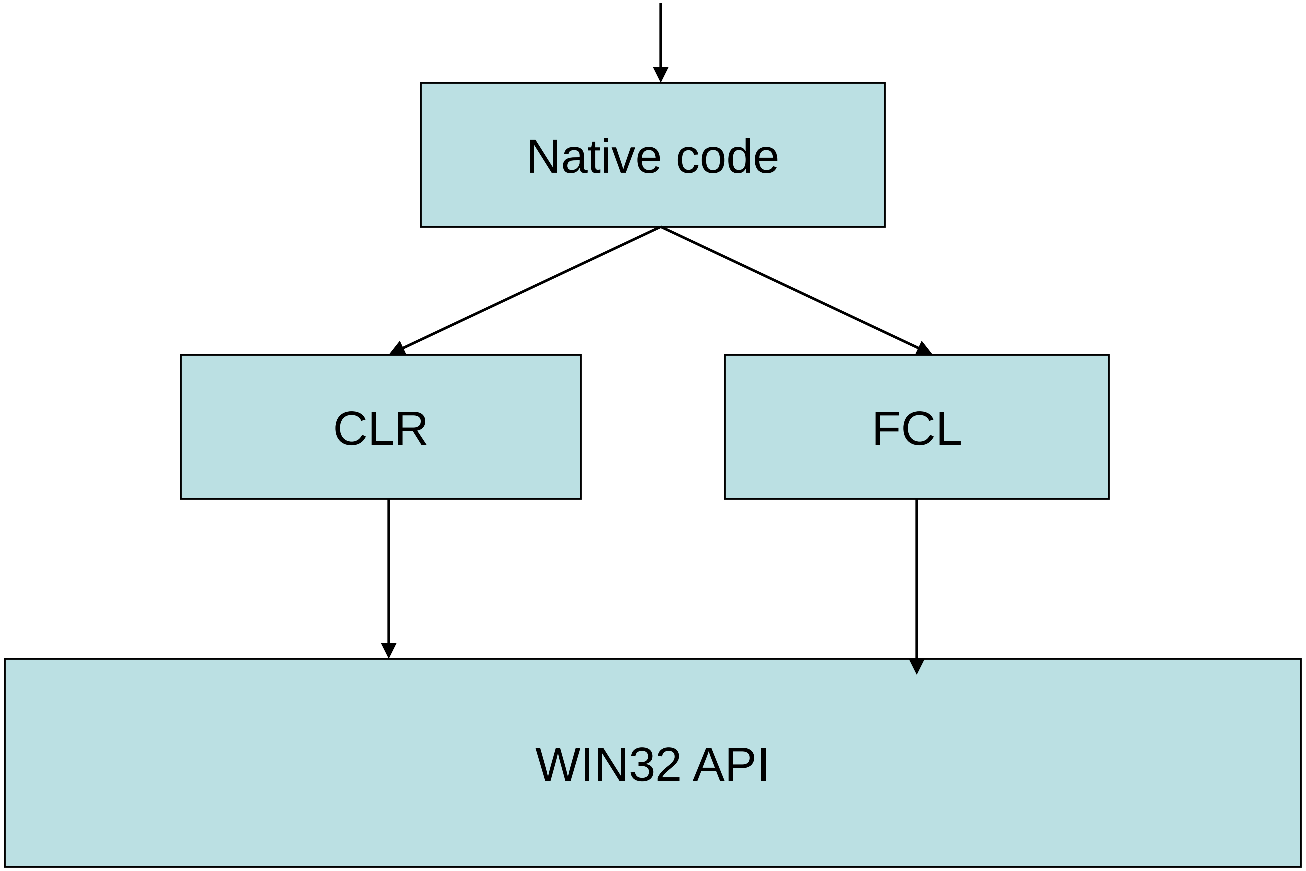
C# compiler

Assembly (.exe)

JIT compiler

Native code

Native code is cached for repetitive use



Native code

CLR

FCL

WIN32 API

# Metadata

- Provides a description of a module
- Includes types defined in the module
  - Classes, structs, enumerations, etc.
- Documents methods
- Fields
- Properties
- External references
- Allows use of a module without type libraries etc.
- ILDASM program can be used to display metadata

# Common Intermediate Language

- Uses a stack based architecture
- Machine independent
- Small instruction set
- *Meta* type operations (directives)

# Assemblies

- A group of one or more files (modules) grouped together
- Fundamental unit of security
- Supports versioning
- Can contain compiled modules from multiple languages
- Contains a *manifest*
- Only assemblies can be executed, not modules

# A Multi-file Assembly

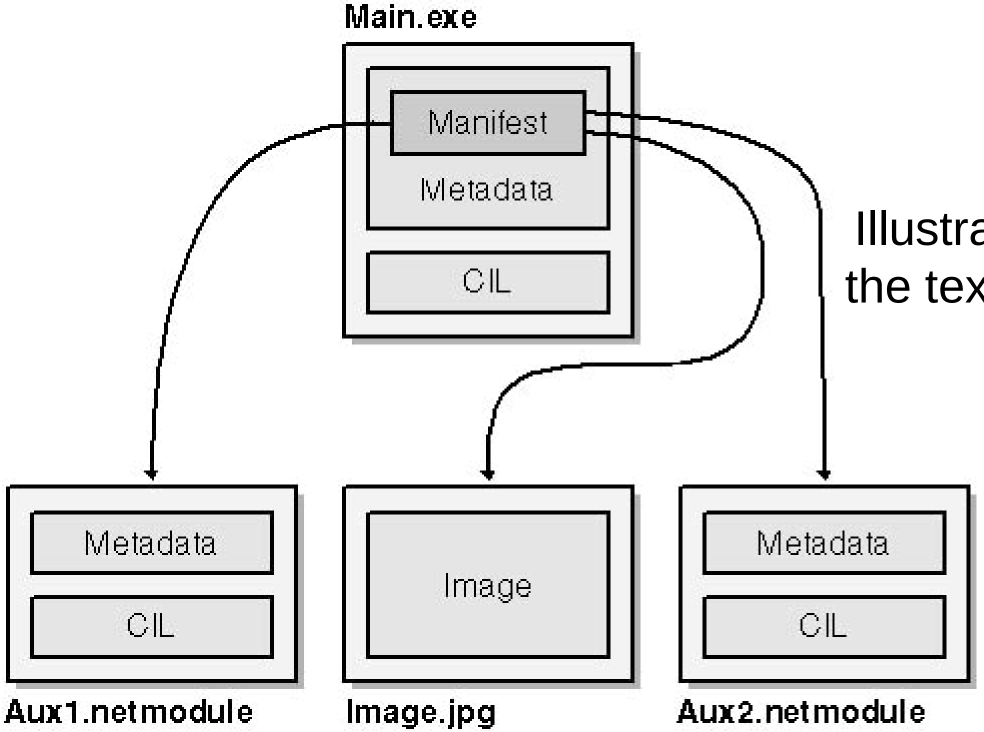


Illustration from the text in MELL

# Framework Class Library

- Replaces use of the WIN32 API
- Includes functionality of MFC
- Includes classes for general use such as container classes
- Over 7,000 types in the library
- Divided into functional groupings using *namespaces*
- Used by all .NET languages
  - Makes switching languages easier

# Hello World in C#

```
using System;
  class MyApp
  {
    static void Main ()
    {
      Console.WriteLine ("Hello, world");
    }
  }
```